

This listing of claims will replace all prior versions, and listings, of claims in the application:

**The Status of the Claims**

1. (Currently Amended) A method of updating code comprising:  
booting an operating system to a runtime environment;  
receiving a pre-boot code update;  
storing the pre-boot code update to a first non-volatile memory if the pre-boot code update fits within an allocated space in the first non-volatile memory;  
setting an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update;  
reading the pre-boot code update;  
implementing the pre-boot code update; and  
clearing the indication that the pre-boot code update is to be implemented.
2. (Original) A method as defined by claim 1, comprising writing the pre-boot code update to a second non-volatile memory if the pre-boot code update does not fit within the allocated space in the first non-volatile memory and writing in the first non-volatile memory a pointer to the pre-boot code update stored in the second non-volatile memory.
3. (Original) A method as defined by claim 2, wherein the second non-volatile memory comprises a portion of a mass storage device.

4. (Original) A method as defined by claim 1, wherein the pre-boot code update comprises a firmware update.
5. (Original) A method as defined by claim 1, wherein the indication that the pre-boot code update is to be implemented comprises setting a flag.
6. (Original) A method as defined by claim 1, wherein the first non-volatile memory comprises a flash memory storing firmware instructions.
7. (Original) A method as defined by claim 1, wherein the pre-boot code update is stored in a host-protected architecture.
8. (Currently Amended) An article of manufacture comprising a machine-accessible medium having a plurality of machine accessible instructions that, when executed, cause a machine to:
  - boot an operating system to a runtime environment;
  - receive a pre-boot code update;
  - store the pre-boot code update to a first non-volatile memory if the pre-boot code update fits within an allocated space in the first non-volatile memory;
  - set an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update;
  - read the pre-boot code update;

implement the pre-boot code update; and  
clear the indication that the pre-boot code update is to be implemented.

9. (Original) A machine-accessible medium as defined by claim 8, wherein the machine accessible instructions, when executed, cause the machine to write the pre-boot code update to a second non-volatile memory if the code update does not fit within the allocated space in the first non-volatile memory and write in the first non-volatile memory a pointer to the pre-boot code update stored in the second non-volatile memory.

10. (Original) A machine-accessible medium as defined by claim 9, wherein the second non-volatile memory comprises a portion of a mass storage device.

11. (Original) A machine-accessible medium as defined by claim 8, wherein the pre-boot code update comprises a firmware update.

12. (Original) A machine-accessible medium as defined by claim 8, wherein the indication that the pre-boot code update is to be implemented comprises setting a flag.

13. (Original) A machine-accessible medium as defined by claim 8, wherein the memory comprises a flash memory storing firmware instructions.

14. (Original) A machine-accessible medium as defined by claim 8, wherein the pre-boot

code update is stored in a host-protected architecture.

15. (Currently Amended) A system comprising:

a network connection;

a flash memory;

a non-volatile memory; and

a processor coupled to the network connection, the flash memory and the non-volatile memory, the processor programmed to:

boot an operating system to a runtime environment;

receive a pre-boot code update;

store the pre-boot code update to the flash memory if the pre-boot code update fits within an allocated space in the flash memory;

set an indication that a pre-boot code update is to be implemented in response to storing the pre-boot code update;

read the pre-boot code update from the flash memory;

implementing the pre-boot code update; and

clearing the indication that the pre-boot code update is to be implemented.

16. (Original) A system as defined by claim 15, wherein the processor is programmed to write the pre-boot code update to the non-volatile memory if the pre-boot code update does not fit within the allocated space in the flash memory and to write in the flash memory a pointer to the pre-boot code update stored in the non-volatile memory.

17. (Original) A system as defined by claim 15, wherein the pre-boot code update comprises a firmware update.
18. (Original) A system as defined by claim 15, wherein the pre-boot code update is stored in a host-protected architecture.
19. (Original) A system as defined by claim 15, wherein the non-volatile memory comprises a portion of a mass storage device.
20. (Original) A system as defined by claim 15, wherein the allocated space in the flash memory comprises a variable storage space.